Topics

- Other Memory Organizations
- Discontinuous Address Spaces
- Avoiding Arithmetic Calculation
- Address Translation or Address Mapping
- Example of a Virtual Memory System
- Interface To Multiple Physical Memory Systems
- Virtual Memory Terminology
- Definition of Virtual Memory
Topics

- Segmentation
- Protection
- Changing the Virtual Space
- Base-Bound Registers
- Technologies to Create Dynamic Virtual Memories
- Multiple Virtual Spaces and Multiprogramming
- Motivation for Virtual Memory
- Other Memory Organizations
- Discontinuous Address Spaces
Summary

Consequences For Programmers

Translation Look-aside Buffer

Storing Page Tables

Control Bits

Using Powers Of Two

Address Translation

Terminology

Page Fault and Replacement

Demand Paging

Topics
This chapter covers:

- Motivation
- Technologies to create virtual address spaces
- Mapping between virtual and physical memory
- How operating system (OS) uses virtual memory

This chapter covers: 

Introduction
Definition of Virtual Memory

• A mechanism that hides details of underlying physical memory to provide a more convenient memory environment.
• A memory access scheme to overcome limitations of physical memory.

Example from previous chapter:

- A controller that provides byte addresses while underlying physical memory uses word addressing choosing powers of two avoids arithmetic computation and makes translations of byte addresses to word address trivial.

An example from previous chapter:

- A memory access scheme to overcome limitations of physical memory.

A memory to provide more convenient memory environment:

- A mechanism that hides details of underlying physical memory.
Virtual Memory Terminology

- Memory Management Unit (MMU)
  - An intelligent memory controller that creates virtual memory

- Real memory = physical memory
- Real address = physical address
- Real address space = physical address space

Real: a synonym for physical, i.e.

- Processor generates virtual addresses
- An intelligent memory controller that creates virtual

Memory Management Unit (MMU)
Architecture in which two dissimilar memories connect to

- Figure shows a 4-byte/word SRAM and 8-byte/word DRAM connected together
- Both physical memories are integrated into a single virtual address space

Example of connecting dissimilar memories

Interface To Multiple Physical Memory Systems
Virtual memory system that divides an address space among two physical memories. MMU uses address to decide which memory to access.

Example of a Virtual Memory System
Address Translation Or Address Mapping

Physical memories.

previous figure. The MMU maps virtual address space onto two
Algorithm used by MMU to create virtual memory shown in

receive memory request from processor;

let A be the address in the request;

if ( A > 1000 ) {
    A = A − 1000;
    pass the modified request to memory 2;
} else {
    pass the unmodified request to memory 1;

}
Address Translation Or Address Mapping

The processor and the addresses used by each memory system must translate between the addresses used by the MMU at zero, the addresses used by each underlying memory system. Because each memory uses address space that start at zero, the illusion of a single, uniform memory can use multiple underlying physical memory systems to provide a processor with the MMU.
Avoiding Arithmetic Calculation

A computation such as subtraction involves arithmetic operations. To decide which memory to access and perform the necessary address correspondence to a power of two, the MMU allows the MMU to choose a virtual address space on a boundary that divides a virtual address space. Now MMU looks at specific bits of the virtual address powers of two. Divide address space along boundaries that correspond to solutions and time. A computation such as subtraction involves both hardware and time.

Dividing a virtual address space on a boundary that corresponds to a power of two allows the MMU to choose a physical memory and perform the necessary address translation without requiring arithmetic operations.
<table>
<thead>
<tr>
<th>Binary Values in Binary</th>
<th>Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000000000</td>
<td>0</td>
</tr>
<tr>
<td>00000000000000</td>
<td>1</td>
</tr>
<tr>
<td>00000000000000</td>
<td>2</td>
</tr>
<tr>
<td>00000000000000</td>
<td>3</td>
</tr>
</tbody>
</table>

Addresses above 1023 are identical except for the highest order bit.

Binary values of addresses in the range 0 to 2047. Note,
What happens if arbitrary amount of memory is installed?

Example, $M_1 = 1024$ and $M_2 = 812$ bytes in the previous example.

Physical memory

virtual addresses 1836-2047 do not correspond to any physical memory

however, some virtual addresses do not correspond to any physical memory

virtual address 1836-2047 do not correspond to any physical memory

each physical memory address has a corresponding virtual address

These are called holes.

Discontinuous Address Spaces
Virtual address space of $N$ bytes mapped into two physical memories. The space is not continuous because only part of each memory is present. If a processor attempts to read or write any address that does not correspond to physical memory, an error results.

A virtual address space can be continuous, in which case the address space can contain non-contiguous, in which case the address space can contain non-contiguous.
Other Memory Organizations

- Interleaving bytes among different physical memory modules
  - Example: 4 bytes of a word interleaved among 4 physical memories

- Advantage: can access the four bytes simultaneously
  - Low-order 2 bits specify 4 the four bytes
  - Interleaving bytes among the four physical memories

Dr. Rajesh Subramanayan, 2005
Motivation for Virtual Memory

- Homogeneous integration of hardware
- Integration under single virtual space (see earlier example)
- Different size, types, word-size, physical memories can be

• Programming convenience
  - Programming and changes to memory access would be difficult for programmer
  - Without virtual addresses, processor will need separate addresses for each physical memory.
Motivation for Virtual Memory

- Support for multiprogramming helps in allowing multiple programs to run at the same time.
- Protection of programs and data.
- CPU uses modes of execution to determine which instruction is allowed at any time.
- Virtual memory linked to protection.
Multiple Virtual Spaces and Multiprogramming

• How to avoid using same memory location by two or more programs running at the same time?

• Use virtual memory to establish a separate virtual address for each program.

• Old method: partitioning, split physical memory into areas

  • advantage, no conflict
  • disadvantage, total area available for a program is a fraction of physical memory

CS250 -- Part III

Dr. Rajesh Subramaniam, 2005
Virtual address spaces mapped onto a single physical memory.
Technologies to Create Dynamic Virtual Memories

- Demand Paging
- Segmentation
- Base-bound registers
Base-Bound Registers

• Creates a single virtual address space and maps the space onto a region of physical memory

- Base register specifies location of virtual space
- Bound register specifies the size of address space

Base and bound are two registers and a part of MMU
Changing the Virtual Space

Virtual memory that uses base-bound mechanism.

A base-bound mechanism uses two values in the MMU to specify how the virtual address space maps onto the physical address space. The base-bound mechanism is powerful because an operating system can change the mapping dynamically.

Virtual memory and physical memory that uses base-bound mechanism.
Protection

A base register can map from virtual to physical address.

This does not prevent a program from accidentally or maliciously referencing large memory locations.

The bound register is used to ensure that programs will not exceed allocated space.

The MMU checks memory reference to bound register values and raises an error if address is too large.

A base register can map from virtual to physical address.
Segmentation refers to a virtual memory scheme in which programs are divided into variable-size blocks, and only the blocks currently needed are kept in memory. Because it leads to a problem known as memory fragmentation, segmentation is seldom used.
Demand Paging

- An alternative to segmentation.
- Scheme is as follows:
  - Divide program into pieces
  - Keep it in external storage till needed
  - Load when needed
- Pieces are fixed-size blocks called pages. Size of pages used to be 512 bytes or 1 KB, nowadays Pentium 4 KB.
- Hardware handles address mapping and detects missing pages.
- Software moves pages between external store and physical memory.
Page Fault and Replacement

- Page Fault: when a reference is made to a missing page
- Page replacement: choosing a page to be removed from physical memory, when it is full and new page needs to be loaded.
- Software algorithms handle page replacement, when it is full and new page needs to be loaded.
- Choosing page to be removed from physical memory,
- Page Replacement

Page Fault and Replacement
Terminology

- Page: a block of program address space
- Page in memory
- Resident
- Frame: a slot of physical memory that can hold a page
- Physical memory
- Unit of memory moved at a time from external storage to
- Block of program address space
- Page
• **Page Table**

  a 1D list indexed by page number. Value is null if page is not resident, and corresponding frame number for pages which are resident in physical memory.
Address Translation

- Assume each page is \( K \) bytes

- Example: translating virtual address \( V \), to corresponding physical address, \( P \)

\[
P = \text{page table}[N] + O
\]

\[
O = C \mod K
\]

- Determine offset within the page

- Corresponding frame number in memory

- Use page number as index into page table to find

\[
N = \text{int}(V/K)
\]

- Determine number of page where address \( V \) lies

- Physical address, \( P \)

Example: translating virtual address \( V \), to corresponding physical address, \( P \), is \( K \) bytes

Assume each page is \( K \) bytes
Using Powers Of Two

Using powers of two eliminates the need for division and remainder computation.

MMU performing address translation in a paging system.
Control Bits

- Presence bit
  - whether page is currently in memory
  - helps detect page fault

- Use bit
  - whether page is currently in memory

- Modify bit
  - set whenever a page is written after it was loaded.
  - page not referenced recently is candidate for eviction
  - provides information for page replacement

- Presence bit
Storing Page Tables

- Page tables are accessed frequently, and determine performance of memory access.
- Page tables can also be quite large.
- Performance of memory access.

Where are page tables stored? Possible locations:

- Special MMU chip external to processor, with high speed
- Hardware interface between processor-MMU
- In the physical memory. SRAM for page table, DRAM
- Subset of page table (TLB) may be stored in high speed memories like CAM
- Storing Page Tables
A part of the physical memory reserved to store page tables.
Translation Look-aside Buffer

- High speed hardware to optimize performance of demand paging
- Provides fast table lookups, TLB is a CAM-like storage device.
- Processor stores page table in memory and subset of page table in TLB. Simultaneously accesses both memory and TLB for page translation. Accepts first reply.
- Processors tend to fetch successive instructions from same page. Recently accessed page table entries are stored in TLB.
Consequences For Programmers

• Programmers can affect virtual memory performance.

• Example 2D array stored in memory:
  - Can be stored as row-major or column-major.
  - All elements of a row stored in contiguous memory.
  - Can be stored as row-major or column-major.
  - Example 2D array stored in memory.
  - Programmers can affect virtual memory performance.

- Ensures contiguous memory are accessed in sequence; all elements of a rows stored in contiguous memory locations.

- Programmer writing nested loop for $A[i,j]$ with row-major storage should have inner loop as $j$, and inner loop as $i$ with column-major storage.

- Results in less page faults in the same page.
Consequences For Programmers

2D array stored in row-major order. A row is contiguous in memory.

<table>
<thead>
<tr>
<th>Row N</th>
<th>Row 0</th>
<th>Row 1</th>
<th>Row 2</th>
<th>Row 3</th>
<th>Row 4</th>
<th>Row 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
Virtual memory system hides details of underlying physical operations. Using powers of two helps avoid arithmetic operations device.

Important subset of page table is stored in TLB, a high speed

Page table to map virtual address to physical address

Demand paging, the most popular among the three uses a virtual technology.

Base-bound, Segmentation, and Demand Paging are three virtual technologies.

Multi-programming and protection

Virtual memory is convenient for programming, and supports memory

Summary